server.R

s435389

2019-07-11

# function derived from the highlightHTMLcells() function of the highlightHTML package  
colortable <- function(htmltab, css, style="table-condensed table-bordered"){  
 tmp <- str\_split(htmltab, "\n")[[1]]  
 CSSid <- gsub("\\{.+", "", css)  
 CSSid <- gsub("^[\\s+]|\\s+$", "", CSSid)  
 CSSidPaste <- gsub("#", "", CSSid)  
 CSSid2 <- paste(" ", CSSid, sep = "")  
 ids <- paste0("<td id='", CSSidPaste, "'")  
 for (i in 1:length(CSSid)) {  
 locations <- grep(CSSid[i], tmp)  
 tmp[locations] <- gsub("<td", ids[i], tmp[locations])  
 tmp[locations] <- gsub(CSSid2[i], "", tmp[locations],  
 fixed = TRUE)  
 }  
 htmltab <- paste(tmp, collapse="\n")  
 Encoding(htmltab) <- "UTF-8"  
 list(  
 tags$style(type="text/css", paste(css, collapse="\n")),  
 tags$script(sprintf(  
 '$( "table" ).addClass( "table %s" );', style  
 )),  
 HTML(htmltab)  
 )  
}  
  
server <- function(input, output, session) {  
  
 # COLORS  
 # #e6550d rgba(230, 85, 13, .5)  
 # #7fcdbb rgba(127, 205, 187, .5)  
 reddish <- 'rgba(230, 85, 13, 1)'  
 blueish <- 'rgba(44, 127, 184, 1)'  
 reddish\_5 <- 'rgba(230, 85, 13, .3)'  
 blueish\_5 <- 'rgba(44, 127, 184, .3)'  
  
  
  
 # ANOVA DATA  
 aov\_dat <- data.frame(cbind(Condition = rep(c("Treatment", "Control"), 2),  
 Gender = rep(c("Female", "Male"), each = 2)),  
 cbind(Mean = c(3, 2, 2, 3)))  
 rownames(aov\_dat) <- paste0("Group\_", 1:nrow(aov\_dat))  
 changed\_aov\_dat <- aov\_dat  
  
  
 # ANCOVA DATA  
 n\_ancova <- 100  
 anco\_endpt <- c(1, 2, 2, 1.5)  
 ancoef <- c(1, -.5)  
  
 set.seed(1321)  
 anco\_dat\_x\_1 <- as.numeric(scale(rnorm(n\_ancova/2)))  
 anco\_dat\_x\_2 <- as.numeric(scale(rnorm(n\_ancova/2)))  
 anco\_dat\_resid\_1 <- as.numeric(scale(rnorm(n\_ancova/2)))  
 anco\_dat\_resid\_2 <- as.numeric(scale(rnorm(n\_ancova/2)))  
  
 anco\_dat\_resid\_1 <- resid(lm(y~x, data = data.frame(x = anco\_dat\_x\_1,  
 y = anco\_dat\_resid\_1)))  
 anco\_dat\_resid\_2 <- resid(lm(y~x, data = data.frame(x = anco\_dat\_x\_2,  
 y = anco\_dat\_resid\_2)))  
 anco\_dat\_g <- rep(0:1, each = n\_ancova/2)  
 anco\_dat\_x <- c(anco\_dat\_x\_1, anco\_dat\_x\_2) \* .15 + .5  
 anco\_dat\_resid <- c(anco\_dat\_resid\_1, anco\_dat\_resid\_2) \* .35  
  
 anco\_dat <- data.frame(g = anco\_dat\_g,  
 x = anco\_dat\_x,  
 y = anco\_endpt[1 + 2 \* anco\_dat\_g] + # Intercept  
 ancoef[1 + anco\_dat\_g] \* anco\_dat\_x + # Regression  
 anco\_dat\_resid) # Residual  
 changed\_anco\_dat <- anco\_dat  
  
  
 svar <- function(x) mean((x - mean(x))^2)  
 compute\_aov <- function(mus, n = 60) {  
  
 MSb <- svar(mus) \* n  
 MSg <- svar(c(sum(mus[1:2]), sum(mus[3:4])) / 2) \* n  
 MSc <- svar(c(sum(mus[c(1, 3)]), sum(mus[c(2, 4)])) / 2) \* n  
 MSint <- MSb - MSg - MSc  
  
 p\_vec <- pf(q = c(MSc, MSg, MSint), df1 = 1, df2 = n - 4, lower.tail = FALSE)  
  
 out <- cbind(c("Condition", "Gender", "Condition:Gender"),  
 sprintf("%.1f", c(MSc, MSg, MSint)),  
 1,  
 n - 4,  
 sprintf("%.3f", p\_vec, 3))  
 colnames(out) <- c(" ", "F-value", "df 1", "df 2", "p-value")  
 out  
 }  
  
 compute\_anco <- function(dat) {  
 coef\_mat <- summary(lm(y ~ I(g == 0) + x, data = dat))$coefficients[-1,]  
 out <- cbind(c("Condition", "Covariate"),  
 sprintf("%.2f", coef\_mat[, 1]),  
 sprintf("%.2f", coef\_mat[, 3]),  
 sprintf("%.3f", coef\_mat[, 4]))  
 colnames(out) <- c(" ", "Coefficient", "t-value", "p-value")  
 out  
 }  
 compute\_anco\_int <- function(dat) {  
 coef\_mat <- summary(lm(y ~ I(g == 0) \* x, data = dat))$coefficients[-1,]  
 out <- cbind(c("Condition", "Covariate", "Condition:Covariate"),  
 sprintf("%.2f", coef\_mat[, 1]),  
 sprintf("%.2f", coef\_mat[, 3]),  
 sprintf("%.3f", coef\_mat[, 4]))  
 colnames(out) <- c(" ", "Coefficient", "t-value", "p-value")  
 print(coef\_mat)  
 out  
 }  
  
  
  
 output$anova\_plot <- highcharter::renderHighchart({  
  
 dropFunction <- JS("function(event){  
 Shiny.onInputChange('drop\_result\_aov', [this.y, this.series.name, this.x]);  
 chart.series[0].data[0].update(y += 10);}")  
  
 condMainDat <- rbind(Control = rep(mean(aov\_dat[aov\_dat$Condition == "Control", "Mean"]), 2),  
 Treatment = rep(mean(aov\_dat[aov\_dat$Condition == "Treatment", "Mean"]), 2))  
  
 highchart() %>%  
 hc\_chart(animation = FALSE) %>%  
 hc\_add\_theme(hc\_theme\_google()) %>%  
 hc\_title(text = "Drag-around ANOVA") %>%  
 hc\_tooltip(valueDecimals = 1) %>%  
 hc\_yAxis(min = 0, max = 5, title = list(text = "Group mean")) %>%  
 hc\_subtitle(text = "2x2 Anova model") %>%  
 hc\_xAxis(categories = c("Female", "Male"), title = list(text = "Gender")) %>%  
 hc\_add\_series(name = "Treatment", data = aov\_dat[aov\_dat$Condition == "Treatment", "Mean"],  
 draggableY = TRUE, color = blueish) %>%  
 hc\_add\_series(name = "Control", data = aov\_dat[aov\_dat$Condition == "Control", "Mean"],  
 draggableY = TRUE, color = reddish) %>%  
 hc\_add\_series(name = "Main Effect Condition", data = condMainDat["Treatment", ],  
 draggableY = FALSE, color = blueish) %>%  
 hc\_add\_series(name = "Main Effect Condition", data = condMainDat["Control", ],  
 draggableY = FALSE, color = reddish) %>%  
 hc\_legend(labelFormatter = JS("function(e) {return this.name;}")) %>%  
 hc\_plotOptions(  
 series = list(  
 point = list(  
 events = list(drag = dropFunction)  
 ),  
 dragPrecisionY = .1,  
 dragMinY = 0,  
 dragMaxY = 5,  
 animation = FALSE,  
 stickyTracking = FALSE  
 ),  
 column = list(  
 stacking = "normal"  
 ),  
 line = list(  
 cursor = "ns-resize"  
 )  
 ) -> h1  
  
 # if (input$showCondition) {  
 #  
 # condMainDat <- cbind(Control = c(1.5, mean(aov\_dat[aov\_dat$Condition == "Control", "Mean"])),  
 # Treatment = c(1.5, mean(aov\_dat[aov\_dat$Condition == "Treatment", "Mean"])))  
 #  
 # print(condMainDat)  
 #  
 # h1 %>% hc\_add\_series(name = "Main Effect Condition",  
 # data = condMainDat,  
 # draggableY = FALSE, color = reddish) -> h1  
 # }  
  
 return(h1)  
 })  
  
  
  
 output$ancova\_plot <- highcharter::renderHighchart({  
  
 dropFunction <- JS("function(event){  
 Shiny.onInputChange('drop\_result\_anco', [this.y, this.series.name, this.x]);}")  
  
 is\_control <- as.logical(changed\_anco\_dat[, 1])  
  
 highchart() %>%  
 hc\_chart(animation = FALSE, type = "line") %>%  
 hc\_add\_theme(hc\_theme\_google()) %>%  
 hc\_title(text = "Drag-around ANCOVA") %>%  
 hc\_tooltip(headerFormat = "", valueDecimals = 2) %>%  
 hc\_yAxis(min = 0, max = 5,  
 title = list(text = "Outcome")  
 ) %>%  
 hc\_xAxis(min = -.2, max = 1.2,  
 title = list(text = "Covariate")) %>%  
 hc\_subtitle(text = "Ancova with two groups (n = 60 for each group)") %>%  
 hc\_legend(labelFormatter = JS("function(e) {return this.name;}")) %>%  
 hc\_add\_series(name = "Treatment", data = anco\_endpt[1:2], type = "line",  
 draggableY = TRUE, color = blueish) %>%  
 hc\_add\_series(name = "Control", data = anco\_endpt[3:4], type = "line",  
 draggableY = TRUE, color = reddish) %>%  
 hc\_add\_series(data = changed\_anco\_dat[!is\_control, 2:3],  
 type = "scatter", color = blueish\_5, showInLegend = FALSE) %>%  
 hc\_add\_series(data = changed\_anco\_dat[is\_control, 2:3],  
 type = "scatter", color = reddish\_5, showInLegend = FALSE) %>%  
 hc\_plotOptions(  
 series = list(  
 point = list(  
 events = list(drop = dropFunction)  
 ),  
 dragPrecisionY = .1,  
 dragPrecisionY = .1,  
 dragMinY = 0,  
 dragMaxY = 5,  
 dragMinX = 0,  
 dragMaxX = 1,  
 animation = FALSE,  
 stickyTracking = FALSE  
 ),  
 column = list(  
 stacking = "normal"  
 ),  
 line = list(  
 cursor = "ns-resize"  
 )  
 ) -> h2  
  
 return(h2)  
 })  
  
  
 makeReactiveBinding("outputText")  
 makeReactiveBinding("changed\_aov\_dat")  
 makeReactiveBinding("changed\_anco\_dat")  
 makeReactiveBinding("ancoef")  
 makeReactiveBinding("anova\_tab")  
 makeReactiveBinding("ancova\_tab")  
 makeReactiveBinding("ancova\_int\_tab")  
  
 # outputText <- "Nothing yet."  
 outputText <- ""  
  
 anova\_tab <- compute\_aov(aov\_dat[, 3])  
 ancova\_tab <- compute\_anco(anco\_dat)  
 ancova\_int\_tab <- compute\_anco\_int(anco\_dat)  
  
  
 observeEvent(input$drop\_result\_aov, {  
 newy <- round(as.numeric(input$drop\_result\_aov[1]), 1)  
 cond <- input$drop\_result\_aov[2]  
 gend <- ifelse(as.numeric(input$drop\_result\_aov[3]), "Male", "Female")  
 # outputText <<- paste0("Hey! You've just moved the mean of ", tolower(gend), "s from the ", tolower(cond),  
 # " condition to ", newy, ".")  
 changed\_aov\_dat[(aov\_dat$Condition == cond) & (aov\_dat$Gender == gend), "Mean"] <<- newy  
  
 anova\_tab <<- compute\_aov(changed\_aov\_dat[, 3], n = input$n\_anova)  
  
 })  
  
 observeEvent(input$n\_anova, {  
 newy <- round(as.numeric(input$drop\_result\_aov[1]), 1)  
 cond <- input$drop\_result\_aov[2]  
 gend <- ifelse(as.numeric(input$drop\_result\_aov[3]), "Male", "Female")  
 changed\_aov\_dat[(aov\_dat$Condition == cond) & (aov\_dat$Gender == gend), "Mean"] <<- newy  
  
 anova\_tab <<- compute\_aov(changed\_aov\_dat[, 3], n = input$n\_anova)  
 })  
  
  
 observeEvent(input$drop\_result\_anco, {  
 newy <- round(as.numeric(input$drop\_result\_anco[1]), 1)  
 cond <- input$drop\_result\_anco[2]  
 is\_end <- as.numeric(input$drop\_result\_anco[3])  
  
 is\_control <- identical(cond, "Control")  
  
 # Update endpoints, then coefficients  
 anco\_endpt[1 + 2\*is\_control + is\_end] <<- newy  
 ancoef[1 + is\_control] <<- anco\_endpt[2 + 2\*is\_control] - anco\_endpt[1 + 2\*is\_control]  
  
 changed\_anco\_dat <<- data.frame(g = anco\_dat\_g,  
 x = anco\_dat\_x,  
 y = anco\_endpt[1 + 2 \* anco\_dat\_g] + # Intercept  
 ancoef[1 + anco\_dat\_g] \* anco\_dat\_x + # Regression  
 anco\_dat\_resid) # Residual  
  
 ancova\_tab <<- compute\_anco(changed\_anco\_dat)  
 ancova\_int\_tab <<- compute\_anco\_int(changed\_anco\_dat)  
 })  
  
 # observeEvent(input$n\_ancova, {  
 # newy <- round(as.numeric(input$drop\_result\_aov[1]), 1)  
 # cond <- input$drop\_result\_aov[2]  
 # gend <- ifelse(as.numeric(input$drop\_result\_aov[3]), "Male", "Female")  
 # # outputText <<- paste0("Hey! You've just moved the mean of ", tolower(gend), "s from the ", tolower(cond),  
 # # " condition to ", newy, ".")  
 # changed\_dat[(dat$Condition == cond) & (dat$Gender == gend), "Mean"] <<- newy  
 #  
 # anova\_tab <<- compute\_aov(changed\_dat[, 3], n = input$n\_anova)  
 # # anova\_tab[]  
 # })  
  
 output$text <- renderText({outputText})  
  
 output$aov\_dattab <- renderTable({  
 cbind(changed\_aov\_dat[, 1:2],  
 Mean = sprintf("%.1f", changed\_aov\_dat[, 3]), n = sprintf("%.0f", input$n\_anova))  
 },  
 digits = c(0, 0, 1, 0))  
  
  
 #### ANOVA RESULTS  
 output$anova\_results <- renderUI({  
  
 # define CSS tags  
 css <- c("#sigcol {background-color: #e6ffb3;}",  
 "#inscol {background-color: #ff9999;}")  
  
 sig <- anova\_tab[,"p-value"] < .05  
  
 anova\_tab\_out <- apply(anova\_tab, 2, function(x) paste(x, ifelse(sig, "#sigcol", "#inscol")))  
  
 # generate html table with pander package and markdown package  
 htmltab <- markdownToHTML(  
 text = pandoc.table.return(  
 anova\_tab\_out,  
 style="rmarkdown", split.tables=Inf  
 ),  
 fragment.only=TRUE  
 )  
 colortable(htmltab, css)  
 })  
  
  
  
 #### ANCOVA RESULTS  
  
 output$ancoef\_tab <- renderTable({  
 data.frame(Group = c("Treatment", "Control"), Intercept = anco\_endpt[c(1, 3)], Coefficient = ancoef)  
 })  
  
  
 output$ancova\_results <- renderUI({  
  
 # define CSS tags  
 css <- c("#sigcol {background-color: #e6ffb3;}",  
 "#inscol {background-color: #ff9999;}")  
  
 sig <- ancova\_tab[,"p-value"] < .05  
  
 ancova\_tab\_out <- apply(ancova\_tab, 2, function(x) paste(x, ifelse(sig, "#sigcol", "#inscol")))  
  
 # generate html table with pander package and markdown package  
 htmltab <- markdownToHTML(  
 text = pandoc.table.return(  
 ancova\_tab\_out,  
 style="rmarkdown", split.tables=Inf  
 ),  
 fragment.only=TRUE  
 )  
 colortable(htmltab, css)  
 })  
  
 output$ancova\_int\_results <- renderUI({  
  
 # define CSS tags  
 css <- c("#sigcol {background-color: #e6ffb3;}",  
 "#inscol {background-color: #ff9999;}")  
  
 sig <- ancova\_int\_tab[,"p-value"] < .05  
  
 ancova\_int\_tab\_out <- apply(ancova\_int\_tab, 2, function(x) paste(x, ifelse(sig, "#sigcol", "#inscol")))  
  
 # generate html table with pander package and markdown package  
 htmltab <- markdownToHTML(  
 text = pandoc.table.return(  
 ancova\_int\_tab\_out,  
 style="rmarkdown", split.tables=Inf  
 ),  
 fragment.only=TRUE  
 )  
 colortable(htmltab, css)  
 })  
  
  
  
}